# Find Subdomains

Finding subdomains is fundamental. The more subdomains you find, the bigger attack surface you have. Which means bigger possibility of success.

For now this seems to be a very comprehensive list of tools to find subdomains. https://blog.bugcrowd.com/discovering-subdomains

Some tools find some stuff, other tools other stuff. So your best bet is to use a few of them together. Don't forget to brute-force recursively!

## Ways to find subdomains

- Brute force
- Search engine responses
- SSL Certificates
- DNS records

## Brute force

## Search engines

Lists subdomains that is has encountered: https://www.virustotal.com

Not sure how this works: https://dnsdumpster.com

### recon-ng

In order to find subdomains we can use the recon-ng framework. It has the same basic structure as metasploit. You can learn more about this tool in the tools-section.

```
recon-ng

use use recon/domains-hosts/

# This will give you a vast amount of alternatives.

show options

set source cnn.com
```

All these subdomains will be saved in `hosts`, which you can access though: `show hosts`

If some of these subdomains are not given IPs automatically you can just run

```
use recon/hosts-hosts/resolve
run
```

And it will resolve all the hosts in the hosts-file.

### Google Dorks

Using google we can also find subdomains.

This will only give us the subdomains of a site.

`site:msn.com -site:www.msn.com`

`site:*.nextcloud.com`

To exclude a specific subdomain you can do this:

`site:*.nextcloud.com -site:help.nextcloud.com`

### subbrute.py

The basic command is like this

`./subbrute.py -p cnn.com`

https://github.com/TheRook/subbrute

### Knock

I haven't tested this yet. https://github.com/guelfoweb/knock

### Reverse DNS-lookup

If you manage to figure out the IP range that the target owns (see section about nmap below). You can see which machines are online. And then you can run a script to find out the domain-addresses of those machines. That way you might find something new.

The text-file onlyIps.txt is a textfile with one IP-address on each line.

```
#!/bin/bash

while read p; do
  echo $p;
  host  $p
done <onlyIps.txt
```

Here are some more tools that can do reverse lookup http://www.cyberciti.biz/faq/how-to-test-or-check-reverse-dns/

### Online tools

```
https://dnsdumpster.com/

https://pentest-tools.com/information-gathering/find-subdomains-of-domain

http://www.intodns.com/

This tool doesn't enumerate subdomains per se. But it hands of a lot of information about domains.
http://www.dnsstuff.com/
```

## Bypassing CloudFlare

```
https://www.ericzhang.me/resolve-cloudflare-ip-leakage/


#This tool can be used to find old IPs. It could mean that the

http://toolbar.netcraft.com/site_report?url=lyst.com
```

### Brute force dictionaries

If you try to brute force the domains it is a good idea to have a good dictionary. That can be found here:

```
Bitquark
https://github.com/bitquark/dnspop

SecList
https://github.com/danielmiessler/SecLists/tree/master/Discovery/DNS
```

## References

https://en.wikipedia.org/wiki/CNAME_record